VLSI / IC Design Canvas

Addendum to Drawing Screen Enhancement Plan

Transistor-Level IC Design, Layout, Simulation & Standard Cell Library

Version: 1.0

Date: February 2026

Status: Planning / Research

1. Executive Summary

This addendum extends the Drawing Screen Enhancement Plan with an eighth canvas: the **VLSI / IC Design Canvas**. Inspired by tools like Microwind, DSCH, Magic VLSI, and the SkyWater SKY130 open-source PDK ecosystem, this canvas brings transistor-level integrated circuit design directly into the embedded IDE. It goes far beyond Microwind by integrating browser-based SPICE simulation (ngspice via WebAssembly), real open-source PDK support, a modern infinite-canvas UX, and AI-assisted circuit generation.

The goal is to let a developer design a custom IC cell, simulate it, verify layout rules, view the cross-section, export SPICE/Verilog/GDSII, and connect it into the broader embedded design workflow — all without leaving the IDE.

2. Microwind Feature Audit: What We're Matching & Surpassing

The following table catalogs every major Microwind feature and our planned disposition: Match (implement equivalent), Enhance (implement better), or Leapfrog (implement something Microwind cannot do).

Microwind Feature	What It Does	Our Plan	Enhancement Details
NMOS/PMOS Generator	Place transistor with W/L in lambda or microns	Enhance	Drag-drop placement with live DRC, visual W/L handles, multi-finger support, parametric sizing panel
Lambda-Based Design Rules	Technology-scaled rules (.rul files)	Enhance	Full SKY130 PDK rules + custom .rul import; real-time DRC overlay as you draw; color-coded violations
Layer Palette	N+diff, P+diff, poly, metal1-6, N-well, contacts, vias	Enhance	Semantic layer groups with toggle visibility; layer transparency slider; active-layer-only editing mode
Contact/Via Generator	Poly-metal, diff-metal, metal-metal contacts	Match	Same macro with grid-aware auto-sizing; stacked via support
Layout Macros (R, L, C, Diode, Pad)	Pre-built parameterized passive components	Enhance	Expanded library: varactors, guard rings, ESD structures, decoupling cells; user-defined macros
Design Rule Checker (DRC)	Batch check, error overlay on layout	Enhance	Real-time incremental DRC (live as you draw); auto-fix suggestions; severity levels; SKY130/custom rules
Analog Simulator (on-layout)	Transient, DC, I-V curves; BSIM4 models	Leapfrog	ngspice via WebAssembly — full SPICE in browser;

			AC/DC/transient/noise/Monte-Carlo; foundry BSIM models
2D Cross-Section View	Slice through layout showing layers	Enhance	Interactive cursor-driven cross-section; animate fabrication steps; show doping profiles
3D Process Viewer	Extruded 3D view of chip	Enhance	Three.js WebGL 3D with orbit/zoom; transparency per layer; measurement tool in 3D
SPICE Netlist Export	Extract .CIR from layout	Enhance	Extract + simulate in one click; also export Verilog, GDSII, LEF/DEF, CIF
Verilog Compile to Layout	DSCH schematic to Verilog to auto-layout	Leapfrog	AI-assisted layout from Verilog/VHDL; Yosys synthesis integration; OpenROAD-style place-and-route
DSCH Schematic Editor	Logic gates, symbols, timing simulation	Leapfrog	Full transistor-level schematic on infinite canvas (React Flow); drag-drop CMOS gates; bi-directional layout sync
Pre-Built Cell Library	Inverter, NAND, NOR, Full Adder, etc.	Leapfrog	SKY130 standard cell library (HD/HS/LP); user cell library; cell characterization; timing/power arcs
Technology Nodes	0.8µm to 14nm FinFET, 3nm nsFET	Match	SKY130 (130nm) as primary; user-importable .rul files for educational nodes; FinFET visual models
MOS I-V Characteristics	Interactive Id/Vd, Id/Vg plots	Enhance	Live parametric sweeps; BSIM4 model comparison; overlay multiple W/L curves
Voltage Color Map (on-layout sim)	Node voltages as color overlay	Enhance	Animated voltage propagation; current density heatmap; power dissipation overlay

3. Canvas #8: VLSI / IC Design Canvas — Detailed Specification

3.1 Three Integrated Sub-Canvases

Unlike Microwind's separate DSCH and Microwind applications, our canvas unifies three views in a single tabbed workspace with live bi-directional synchronization:

1. **Schematic Canvas** — Transistor-level circuit design (replaces DSCH). Place NMOS/PMOS symbols, passive components, and logic gates. Wire them together with smart routing. Simulate with timing diagrams.
2. **Layout Canvas** — Physical mask layout editor (replaces Microwind). Draw on silicon layers with design rule awareness. Place transistors with specified W/L, add contacts, route metals.
3. **Simulation Dashboard** — Waveform viewer, I-V curve plotter, and analysis results panel. ngspice WebAssembly runs SPICE simulations entirely in-browser with zero server dependency.

Changes in any sub-canvas propagate to the others. Edit a transistor width in the schematic and the layout updates. Move a transistor in layout and the schematic reflects the new connectivity.

3.2 Schematic Canvas (Transistor-Level)

3.2.1 Component Library

The component palette provides drag-and-drop access to:

- **Primitive Devices:** NMOS (4-terminal with bulk), PMOS (4-terminal with bulk), NPN/PNP BJT, Diode, Resistor, Capacitor, Inductor, Varactor
- **Logic Gates:** Inverter, Buffer, NAND2/3/4, NOR2/3/4, AND, OR, XOR, XNOR, Transmission Gate, Tri-State Buffer
- **Sequential Elements:** D Flip-Flop, SR Latch, JK Flip-Flop, D Latch, Register (parameterized width)
- **Arithmetic Blocks:** Half Adder, Full Adder, 4-bit Ripple Adder, Carry-Lookahead Adder, Comparator, Multiplexer (2:1, 4:1, 8:1)
- **Memory Elements:** 6T SRAM Cell, DRAM Cell, ROM Cell, Register File
- **Analog Blocks:** Differential Pair, Current Mirror, Cascode, Op-Amp (basic 2-stage), Bandgap Reference, LDO (basic)
- **Sources & Probes:** VDD/VSS supply, Voltage/Current source (DC, Pulse, Sine, PWL), Voltage Probe, Current Probe, Ground
- **User-Defined Symbols:** Any schematic can be encapsulated as a reusable symbol (like Microwind's Schema-to-Symbol feature)

3.2.2 Transistor Properties Panel

Clicking any MOS transistor opens a properties panel with:

- **Dimensions:** W (width) and L (length) in μm or nm, with technology-minimum enforcement
- **Fingers:** Number of gate fingers (multi-finger layout for wide transistors)

- **Model:** Select BSIM4, BSIM3v3, or Level 1 (educational); auto-populated from PDK
- **Quick I-V Preview:** Inline Id-Vgs and Id-Vds curves for the current W/L at a glance
- **Operating Point:** After simulation, shows Vgs, Vds, Vth, gm, gds, Cgs, Cgd for each device

3.2.3 Simulation Integration

The schematic canvas generates a SPICE netlist automatically. Pressing **Simulate** (or Ctrl+Enter) sends the netlist to **ngspice compiled to WebAssembly**, which runs entirely in the browser. Supported analyses:

- **Transient (.tran):** Time-domain simulation with automatic timestep control
- **DC Operating Point (.op):** Bias point calculation with annotated node voltages
- **DC Sweep (.dc):** Parameter sweep for transfer characteristics and I-V curves
- **AC Analysis (.ac):** Frequency response with Bode magnitude/phase plots
- **Noise Analysis (.noise):** Input/output referred noise spectral density
- **Monte Carlo:** Statistical process variation analysis (mismatch, corners)
- **PVT Corners:** Fast-Fast, Slow-Slow, Typical-Typical with SKY130 corner models

3.3 Layout Canvas (Physical Design)

3.3.1 Layer System

The layout canvas provides a full mask-layer editing environment. For SKY130 technology:

Layer	Color	GDS#	Purpose	Key Rules
N-Well	■ Yellow	64/20	PMOS body	min W: 0.84µm
Active (Diffusion)	■ Green	65/20	Source/Drain	min W: 0.15µm
Polysilicon	■ Red	66/20	Gate	min W: 0.15µm
Local Interconnect	■ Blue	67/20	LI routing	min W: 0.17µm
Metal 1–5	■ Cyan→Purple	68–72	Interconnect	varies per layer
N+/P+ Implant	■ Orange/Pink	93/94	Doping type	encloses active
Contact / Via	■ Black	various	Layer connections	min 0.17×0.17

3.3.2 Drawing Tools

Each layer has dedicated drawing tools, all operating on the infinite canvas with snap-to-grid:

- **Rectangle Tool:** Draw boxes on any layer. Dimensions shown live in µm. Constrain to minimum width.
- **Polygon Tool:** Arbitrary Manhattan (90°) or 45° shapes for custom geometries.
- **Path/Wire Tool:** Draw metal routes with automatic width enforcement and via insertion at layer crossings.
- **MOS Generator:** Click to place NMOS or PMOS. Specify W, L, number of fingers, number of contacts. The tool auto-generates the complete transistor layout including wells, implants, contacts, and substrate ties.
- **Contact Array Generator:** Specify NxM contacts between any two adjacent layers.
- **Guard Ring Generator:** One-click substrate or well guard ring around a region.
- **Smart Ruler:** Measure distances between any two points with DRC-aware feedback (green = passes, red = violation).

3.3.3 Real-Time DRC (Design Rule Checking)

Unlike Microwind's batch DRC, our system performs **incremental real-time DRC** as you draw. Violations appear as red overlays with human-readable messages (e.g., "Metal1 spacing: 0.12µm found, 0.14µm required"). The DRC engine supports:

- Width, spacing, enclosure, extension, and area rules
- Density rules (metal fill requirements)
- Antenna rules (charge accumulation during fabrication)
- Custom rule files (.rul format compatible with Microwind; .lydrc for KLayout format)
- **Auto-Fix:** Right-click a violation to see suggested fixes (e.g., "Expand N-Well by 0.06µm") and apply with one click.

3.3.4 Extraction & LVS

The layout canvas can extract a SPICE netlist from the physical geometry (parasitic extraction). It then compares this against the schematic netlist for Layout vs. Schematic (LVS) verification. Mismatches are highlighted in both canvases simultaneously.

3.3.5 Cross-Section & 3D Views

A draggable cross-section cursor on the layout shows a real-time 2D slice of the chip stack, displaying each layer's material, doping profile, and oxide thickness. The 3D view (Three.js/WebGL) renders the full chip with per-layer transparency control, orbit camera, and measurement tools.

4. Standard Cell Library System

This is where we go far beyond Microwind. Instead of just a handful of pre-built cells, we provide a complete standard cell library ecosystem.

4.1 Bundled Libraries

Library	Source	Cell Count	Use Case
sky130_fd_sc_hd	SkyWater/Google	~430 cells	High-Density: default for digital designs
sky130_fd_sc_hs	SkyWater/Google	~380 cells	High-Speed: performance-critical paths
sky130_fd_sc_lp	SkyWater/Google	~300 cells	Low-Power: battery/IoT applications
Educational (Lambda)	Built-in	~60 cells	Simplified lambda-rule cells for learning (like Microwind)
User Custom	User-created	Unlimited	Save any layout as a reusable library cell

4.2 Cell Browser & Viewer

A sidebar cell browser lets users search and preview any cell before placing it. Each cell shows: schematic symbol, layout thumbnail, truth table (for logic), timing arcs (setup/hold/delay), and power data. Double-clicking opens the cell for inspection or editing.

4.3 Cell Characterization

For user-created cells, an automated characterization flow measures: **propagation delay** (rise/fall for each input-to-output arc), **setup/hold times** (for sequential cells), **power consumption** (dynamic + leakage), and **input capacitance**. Results are exported in Liberty (.lib) format for use in synthesis tools like Yosys.

5. Simulation Engine: ngspice in the Browser

The simulation backbone is ngspice compiled to WebAssembly via Emscripten. This is a proven approach (EEcircuit, ngspiceX) that runs the full ngspice engine locally in the browser with no server round-trip.

5.1 Architecture

- **Engine:** ngspice 43+ compiled with Emscripten to .wasm (~3MB); loaded once, cached by browser
- **Worker Thread:** Simulation runs in a Web Worker to keep the UI responsive during long simulations
- **Plotting:** WebGL-accelerated waveform rendering (webgl-plot or custom PixiJS); handles millions of data points

- **Models:** Bundled BSIM4 model cards for SKY130 (TT/FF/SS/SF/FS corners at -40°C, 27°C, 100°C)
- **Privacy:** All simulation runs locally. No netlist or results are ever uploaded to any server.

5.2 Interactive Features Beyond Microwind

1. **Live Voltage Annotation:** After DC simulation, node voltages appear on the schematic with color coding (red = VDD, blue = VSS, gradient in between). Current flow arrows show direction and magnitude.
2. **Parametric Sweep UI:** Graphical slider to sweep any parameter (W, L, temperature, supply voltage) and watch waveforms update in real-time.
3. **Corner Analysis Dashboard:** Run all PVT corners simultaneously and view results overlaid; highlight worst-case paths.
4. **Power Heatmap:** Overlay power dissipation on the layout during transient simulation. Identify hot spots.
5. **Timing Measurement:** Click any two waveform points to measure propagation delay, rise/fall time, overshoot, and settling time.

6. AI-Assisted IC Design

This is our biggest leap beyond Microwind and all existing educational VLSI tools. The AI assistant integrates directly into the VLSI canvas:

1. **Natural Language to Schematic:** Describe a circuit in words ("2-input CMOS NAND gate with Wp=2µm, Wn=1µm, L=130nm") and the AI generates the transistor-level schematic.
2. **Schematic to Layout:** AI generates an initial layout from the schematic, respecting DRC rules and optimizing for area. The user can then manually refine.
3. **Verilog/VHDL to Transistors:** Given an RTL description, the AI synthesizes to gates (via Yosys) then maps to transistor-level implementation.
4. **Design Review:** AI analyzes a layout and suggests improvements: better transistor sizing for speed/power, missing substrate ties, DRC-risky areas.
5. **Simulation Coach:** AI explains simulation results, identifies problems (e.g., "Your inverter's Vth is 0.95V instead of 0.9V because the PMOS is undersized"), and suggests fixes.
6. **Circuit Tutoring Mode:** Step-by-step guided exercises: "Build a CMOS inverter from scratch," "Design a 2-input NAND," "Create a current mirror." Each step validates before proceeding.

7. Export Formats & Interoperability

Format	Description	Direction	Interop With
SPICE (.cir, .sp)	Circuit netlist	Import + Export	ngspice, LTSpice, HSPICE, PSpice
Verilog (.v)	RTL / gate-level	Import + Export	Yosys, Icarus, Verilator, ModelSim
GDSII (.gds)	Layout masks	Import + Export	KLayout, Magic, Cadence, Synopsys
LEF/DEF	Abstract/placement	Export	OpenROAD, Innovus, ICC2
CIF	Layout interchange	Import + Export	Magic VLSI, older tools
Liberty (.lib)	Timing/power models	Export	Yosys, OpenSTA, Genus
MSK (.msk)	Microwind layout	Import	Microwind (legacy project import)
SVG/PNG/PDF	Visual export	Export	Documentation, papers, presentations
SKY130 PDK	Process design kit	Import	Google/SkyWater open-source foundry flow

8. Cross-Canvas Integration with Existing Seven Canvases

The VLSI/IC Design Canvas connects bidirectionally with the seven canvases from the main plan:

- **VLSI → Pin-Mux Canvas:** A custom IO pad cell in the VLSI canvas maps to a pin on the chip package in the Pin-Mux view.
- **VLSI → Schematic Viewer:** The transistor-level schematic can be viewed alongside the system-level KiCad schematic, showing how custom IC cells connect to the board.
- **VLSI → Waveform Canvas:** Simulation results from ngspice display in the shared waveform viewer alongside logic analyzer captures from real hardware.
- **VLSI → Memory Map:** ROM/SRAM cells designed in VLSI canvas appear in the memory map with their address ranges.
- **VLSI → Register Editor:** Custom peripheral registers designed at transistor level are reflected in the register bit-field editor.
- **VLSI → Block Diagram:** Any custom IC block appears as a black-box symbol in the system block diagram with its ports.

9. Implementation Roadmap

This roadmap adds 20 weeks to the main plan, interleaved with existing sprints:

Sprint	Duration	Deliverables	Dependencies
V1	4 weeks	Layout canvas core: layer system, rectangle/polygon tools, grid, zoom/pan, SKY130 layer palette	Canvas shell from S1
V2	3 weeks	MOS generator, contact generator, DRC engine with SKY130 rules, real-time violation overlay	V1
V3	3 weeks	Schematic canvas: NMOS/PMOS symbols, wire routing, component library (basic gates), netlist generation	React Flow from S4
V4	3 weeks	ngspice WASM integration: compile, netlist handoff, transient/DC/AC analysis, waveform plotting	V3
V5	2 weeks	Schematic-layout synchronization: bi-directional mapping, LVS checking, extraction	V2, V3
V6	2 weeks	Standard cell library browser: SKY130 HD import, cell viewer, search, place from library	V2
V7	2 weeks	2D cross-section, 3D view (Three.js), voltage color map, power heatmap	V4

V8	1 week	Export: GDSII, CIF, SPICE, Verilog, SVG/PNG; Microwind .msk import	V5

Total: **20 weeks** (5 months). MVP with basic layout + simulation at V4 (13 weeks). Full feature set at V8.

10. Technology Comparison: Us vs. Alternatives

Feature	Microwind	Magic VLSI	KLayout	EEcircuit	Ours
Runs in Browser	No (Win)	No (X11)	No (Qt)	Yes	Yes
Integrated Schematic	DSCH (separate)	No	Via xschem	No	Unified canvas
SPICE Simulation	Built-in (basic)	Via ngspice	No	ngspice WASM	ngspice WASM
Real-Time DRC	Batch only	Incremental	Batch	No	Incremental
Open-Source PDK	Educational .rul	SKY130	SKY130	Models only	SKY130 full
Standard Cell Library	~20 cells	Via OpenROAD	Viewer only	No	430+ (SKY130)
AI Assistance	No	No	No	No	Full AI copilot
3D View	Basic	No	Yes	No	WebGL (Three.js)
GDSII Export	CIF only	Yes	Yes	No	Yes
Embedded IDE Integration	No	No	No	No	Full (8 canvases)

11. Success Metrics

1. **CMOS Inverter in <2 minutes:** A student can design, lay out, and simulate a CMOS inverter from scratch in under 2 minutes (vs. ~8 minutes in Microwind).
2. **Full Adder in <10 minutes:** Complete transistor-level design with DRC-clean layout and verified simulation.
3. **Zero DRC violations on first compile:** AI-assisted layout produces DRC-clean results >80% of the time.
4. **Simulation parity:** ngspice WASM results within 0.1% of desktop ngspice for all SKY130 model cards.
5. **60fps layout editing:** Smooth pan/zoom/draw with 1000+ rectangles on the canvas.
6. **Full GDSII round-trip:** Export from our tool, import in KLayout/Magic, and back — zero data loss.

12. Immediate Next Steps

- **POC 1:** Compile ngspice to WebAssembly; run a CMOS inverter transient simulation in a VSCode Webview; plot results with WebGL.
- **POC 2:** Build a minimal layout canvas (HTML5 Canvas + React) with polysilicon, diffusion, and metal1 layers; implement NMOS generator with W/L parameters.
- **POC 3:** Import SKY130 HD standard cells (GDSII) and render them in the layout canvas; verify layer colors and geometry.
- **POC 4:** Implement real-time DRC for 5 basic SKY130 rules (poly width, metal1 spacing, contact size, active enclosure, N-well width).
- **User Testing:** Give 5 VLSI students the tool and time them building an inverter, NAND, and full adder vs. Microwind. Target: 50% faster.